

控制与决策

Control and Decision

混合改进策略的黑猩猩优化算法及其机械应用

何庆, 罗仕杭

引用本文:

何庆, 罗仕杭. 混合改进策略的黑猩猩优化算法及其机械应用[J]. 控制与决策, 2023, 38(2): 354–364.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.1108>

您可能感兴趣的其他文章

Articles you may be interested in

[多策略融合的改进麻雀搜索算法及其应用](#)

Improved sparrow search algorithm with multi-strategy integration and its application

控制与决策. 2022, 37(1): 87–96 <https://doi.org/10.13195/j.kzyjc.2021.0582>

[基于改进量子粒子群的K-means聚类算法及其应用](#)

K-means clustering algorithm based on improved quantum particle swarm optimization and its application

控制与决策. 2022, 37(4): 839–850 <https://doi.org/10.13195/j.kzyjc.2020.1302>

[嵌入Circle映射和逐维小孔成像反向学习的鲸鱼优化算法](#)

Whale optimization algorithm for embedded Circle mapping and one-dimensional oppositional learning based small hole imaging

控制与决策. 2021, 36(5): 1173–1180 <https://doi.org/10.13195/j.kzyjc.2019.1362>

[基于自适应正态云模型的灰狼优化算法](#)

Grey wolf optimization algorithm based on adaptive normal cloud model

控制与决策. 2021, 36(10): 2562–2568 <https://doi.org/10.13195/j.kzyjc.2020.0233>

[混合柯西变异和均匀分布的蝗虫优化算法](#)

Hybrid Cauchy mutation and uniform distribution of grasshopper optimization algorithm

控制与决策. 2021, 36(7): 1558–1568 <https://doi.org/10.13195/j.kzyjc.2019.1609>

混合改进策略的黑猩猩优化算法及其机械应用

何庆[†], 罗仕杭

(贵州大学 大数据与信息工程学院, 贵阳 550025)

摘要: 针对黑猩猩优化算法存在易陷入局部最优、收敛速度慢、寻优精度低等缺陷, 提出混合改进策略的黑猩猩优化算法(SLWChOA). 首先, 利用 Sobol 序列初始化种群, 增加种群的随机性和多样性, 为算法全局寻优奠定基础; 其次, 引入基于凸透镜成像的反向学习策略, 将其应用到当前最优个体上产生新的个体, 提高算法的收敛精度和速度; 同时, 将水波动态自适应因子添加到攻击者位置更新处, 增强算法跳出局部最优的能力; 最后, 通过 10 个基准测试函数、Wilcoxon 秩和检验以及部分 CEC2014 函数进行仿真实验来评价改进算法的寻优性能, 实验结果表明, 所提算法在寻优精度、收敛速度和鲁棒性上均较对比算法有较大提升. 另外, 通过一个机械优化设计实验进行测试分析, 进一步验证了 SLWChOA 的可行性和适用性.

关键词: 黑猩猩优化算法; Sobol 序列; 凸透镜成像; 水波动态自适应因子; 机械优化设计

中图分类号: TP301

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.1108

开放科学(资源服务)标识码(OSID):



引用格式: 何庆, 罗仕杭. 混合改进策略的黑猩猩优化算法及其机械应用[J]. 控制与决策, 2023, 38(2): 354-364.

Chimp optimization algorithm based on hybrid improvement strategy and its mechanical application

HE Qing[†], LUO Shi-hang

(School of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

Abstract: Chimp optimization algorithms (ChOA) based on the convex lens imaging strategy is proposed to overcome the drawbacks of easily trapping into local optimum, slow convergence speed and low optimization precision. Firstly, the population is initialized by the Sobol sequence, which can increase the randomness and diversity of the population, and lay the foundation for the global optimization of the algorithm. Then, the opposition-based learning strategy based on convex lens imaging is introduced, which is applied it to the current optimal individual to generate new individuals, and improve the convergence accuracy and speed of the algorithm. At the same time, the water wave dynamic adaptive factor is added to the attacker's location update to enhance the ability of the algorithm to escape from the local optimum. Finally, The simulation experiments are conducted on the 10 benchmark functions, the Wilcoxon rank sum test and some parts of CEC2014 functions to evaluate the optimization performance of the improved algorithm. The experimental results show that the proposed algorithm has more significant improvement in optimization accuracy, convergence speed and robustness than the comparison algorithm. In addition, three mechanical optimization design experiments are conducted to test and analyze the feasibility and applicability of the improved algorithm.

Keywords: chimp optimization algorithm (ChOA); Sobol sequence; convex lens imaging; water wave dynamic adaptive factor; mechanical optimization design

0 引言

在工程应用和科学研究中存在着大量高维、非线性、目标函数不可导的复杂优化问题, 元启发式算法具有参数少、易操作、不需要梯度信息等优点, 可以很好地在合理的时间且高度复杂的约束下找到这些问题的最优解. 因此, 近年来元启发式算法在生产

调度^[1]、速度预测^[2]、故障诊断^[3]等领域得到了广泛的研究和应用.

黑猩猩优化算法 (ChOA) 是由 Khishe 等^[4] 于 2020 年提出的一种新型元启发式优化算法. 它源于对自然界中黑猩猩个体智力、性动机及捕食行为的模拟, 通过驱赶、追逐、攻击等过程, 构建一种有效

收稿日期: 2021-06-26; 录用日期: 2021-10-27.

基金项目: 国家自然科学基金项目(62166006); 贵州省科学技术厅项目(黔科合基础-ZK[2021] 一般335).

责任编辑: 林崇.

[†]通讯作者. E-mail: qhe@gzu.edu.cn.

的优化方案.虽然ChOA具有原理简单、易操作、需调节参数少等特点,但是ChOA与其他群体智能优化算法相似,存在求解精度低、收敛速度慢和易陷入局部极值等不足.为改善上述缺陷,Ning等^[5]针对BOA(butterfly optimization algorithm)收敛精度低的问题,在个体位置更新处添加自适应余切权重系数,使算法收敛速度和精度得到提高;Long等^[6]提出了一种新的参数 C 策略,以达到平衡GWO(grey wolf optimizer algorithm)的探索 and 开发能力的效果;Kaur等^[7]将正余弦函数引入ChOA个体位置更新处,提高了算法的收敛性能;Jia等^[8]将多项式变异用于ChOA种群初始化,提高了种群的多样性和初始阶段解的质量.

虽然上述文献中的改进策略在一定程度上提高了算法收敛精度和速度,但算法仍存在易陷入局部极值、“早熟”收敛等问题.为此,本文提出混合改进策略的黑猩猩优化算法(SLWChOA).首先,采用Sobol序列产生均匀分布的种群,增加初始种群的多样性和随机性并提高求解效率;其次,引入基于凸透镜成像原理的反向学习策略,将产生的解与当前最优解结合生成更优解,增强群体之间信息交流和学习,提高算法的寻优性能;最后,在攻击者个体位置更新处加入水波动态自适应因子,扩大群体的寻优范围,降低算法陷入局部最优的概率.通过对10个基准测试函数及其Wilcoxon秩和进行检验,并对部分CEC2014函数进行仿真实验并将其应用于1个实际工程问题来验证SLWChOA算法的有效性和可行性.

1 黑猩猩优化算法

ChOA的灵感来源于自然界中黑猩猩的狩猎行为,黑猩猩根据分工采取不同的行动寻找猎物.标准的ChOA算法将黑猩猩群体分为4种类型:攻击者、障碍者、驱赶者和追逐者,其中攻击者是种群的领导者,其他3类黑猩猩协助狩猎,社会地位依次下降.黑猩猩驱逐和追逐猎物的数学模型如下:

$$d = |C \cdot X_{\text{prey}}(t) - m \cdot X_{\text{chimp}}(t)|, \quad (1)$$

$$X_{\text{chimp}}(t+1) = X_{\text{prey}}(t) - a \cdot d. \quad (2)$$

其中: t 为当前迭代次数; X_{prey} 为猎物位置向量; X_{chimp} 为当前黑猩猩位置向量; a 、 m 、 C 为系数向量,计算公式如下:

$$a = 2f \cdot r_1 - f, \quad (3)$$

$$m = \text{Chaotic_value}, \quad (4)$$

$$C = 2 \cdot r_2. \quad (5)$$

这里: r_1 和 r_2 分别为 $[0, 1]$ 间的随机向量; f 为收敛因子,其值随迭代次数增加从2.5非线性减小到0; a 为

决定黑猩猩与猎物距离的随机向量,其值为 $[-f, f]$ 之间的随机数; m 为混沌映射矢量,代表黑猩猩在狩猎过程中性动机的影响; C 为黑猩猩驱逐和追赶猎物的控制系数,其值为 $[0, 2]$ 之间的随机数.

种群初始化后,依次选出4个最优解作为攻击者、障碍者、驱赶者和追逐者的位置,种群中的其他黑猩猩位置围绕以下4种黑猩猩的位置进行更新,其数学模型描述如下:

$$X_1 = X_{\text{attacker}} - a_1 \cdot |C_1 \cdot X_{\text{attacker}} - m_1 \cdot X|. \quad (6)$$

$$X_2 = X_{\text{barrier}} - a_2 \cdot |C_2 \cdot X_{\text{barrier}} - m_2 \cdot X|, \quad (7a)$$

$$X_3 = X_{\text{chaser}} - a_3 \cdot |C_3 \cdot X_{\text{chaser}} - m_3 \cdot X|, \quad (7b)$$

$$X_4 = X_{\text{driver}} - a_4 \cdot |C_4 \cdot X_{\text{driver}} - m_4 \cdot X|. \quad (7c)$$

$$X(t+1) = (X_1 + X_2 + X_3 + X_4)/4. \quad (8)$$

其中: X 为当前黑猩猩的位置向量, X_{attacker} 为攻击者的位置向量, X_{barrier} 为障碍者的位置向量, X_{chaser} 为追逐者的位置向量, X_{driver} 为驱赶者的位置向量, $X(t+1)$ 为当前黑猩猩更新后的位置向量.

2 改进的黑猩猩优化算法

首先,在标准ChOA中,种群初始化采用随机分布的方式,这种方式造成种群多样性差,导致个体前期搜索存在一定的盲目性,使算法收敛速度慢;其次,标准ChOA通过对种群历史前4个最优位置的加权记忆,实现个体在搜索空间快速寻优,但这一做法并未考虑到每个黑猩猩个体自身的搜索经验,导致算法寻优精度低;最后,标准ChOA的种群围绕攻击者、障碍者、驱赶者和追逐者进行位置更新,当这4个个体陷入局部空间极值时,种群会随之陷入局部最优,导致算法出现停滞搜索现象.

综上所述,本文针对上述标准ChOA原理的缺陷,引入对应的策略进行改进.首先,本文通过Sobol序列初始化种群,使种群分布更加均匀,降低前期个体搜索的盲目性,从而提高算法的收敛速度;其次,引用基于凸透镜成像的反向学习策略,通过对算法中当前最优个体进行凸透镜成像反向学习,将各维度的值映射到搜索空间中得到反向解,引导群体向全局最优解附近靠拢,提高算法的寻优精度;最后,在种群的引导者“攻击者”位置更新处添加水波动态自适应因子,利用水波的动态变化的方向和大小不确定性帮助攻击者引导种群在更广阔的范围搜索,增加算法寻优位置的多样性,避免趋同性,降低算法陷入局部最优的概率.

2.1 Sobol序列初始化种群

初始解在解空间的分布情况很大程度上影响元启发式算法的收敛速度和寻优精度. 均匀分布的种群初始解可以有效提高算法的搜索效率. 标准ChOA采用随机方式初始化种群, 这种方法遍历性低且种群分布不均匀. 本文采用Sobol序列初始化种群, Sobol序列是一种低差异序列, 使用确定性拟随机数序列代替伪随机数序列的方法, 将尽可能均匀的点填充至多维超立方体. 因此, Sobol序列在解决概率问题时, 计算效率更高, 采样点覆盖率更广^[9].

设全局解的取值范围为 $[lb, ub]$, 由Sobol序列产生的第 i 个随机数 $S_i \in [0, 1]$, 则种群初始位置表示为

$$X_n = lb + S_i \cdot (ub - lb). \quad (9)$$

假设搜索空间为二维, 上下界分别为0和1, 种群规模为100, 对比随机初始化种群空间分布和Sobol序列初始化种群空间分布如图1所示.

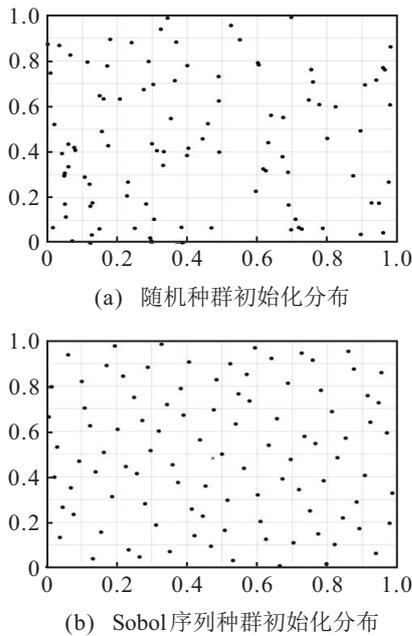


图1 不同方法产生的初始化种群分布

由图1可知, 通过Sobol序列产生的初始种群分布更加均匀, 每个坐标区间都有个体分布其中, 对解空间的覆盖率更广.

2.2 基于凸透镜成像的反向学习策略

由式(6)和(7)可知, 种群中其他黑猩猩个体通过决策层个体(攻击者、障碍者、驱赶者和追逐者)的引导向最优解区域收敛. 然而, 当攻击者、障碍者、驱赶者和追逐者搜索到更好的猎物时, 整个种群会聚集在这4类黑猩猩周围的搜索区域, 导致种群多样性减少, 从而使算法出现“早熟”收敛, 寻优精度降低. 为解决这一问题, 本文提出一种基于凸透镜成像原理的反向学习策略, 将其产生的反向个体与当前最优个体结合

生成新的个体.

黑猩猩在空间中寻找反向点的过程可以近似为凸透镜成像的过程. 假设在一维空间中, 在坐标轴区间 $[lb, ub]$ 上有一个高度为 h 的个体 P , 其在 X 轴的投影为 X (X 为全局最优个体), 基点位置 O (本文取基点位置为 $(lb + ub/2)$) 上放置焦距为 F 的凸透镜, 个体 P 通过凸透镜得到一个高度为 h^* 的倒像 P^* , 此时通过透镜成像在 X 轴上产生第1个反向个体 X^* , 成像原理如图2所示.

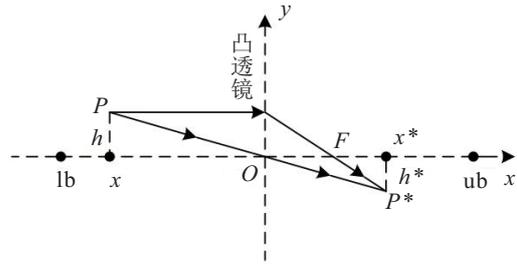


图2 基于凸透镜成像反向学习示意图

图2中, 全局最优个体 X 以 O 为基点找到其对应的反向个体 X^* , 则根据凸透镜成像原理可以得出

$$\frac{(ub + lb)/2 - X}{X^* - (ub + lb)/2} = \frac{h}{h^*}. \quad (10)$$

设 $h/h^* = \eta$ 表示伸缩因子与物和像之间的对应比例关系, 其中 η 称为伸缩因子. 对式(10)进行变化即可得到反向点 X^* 的计算公式

$$X^* = \frac{ub + lb}{2} + \frac{ub + lb}{2 \cdot \eta} - \frac{X}{\eta}. \quad (11)$$

在SLWChOA搜索解的过程中, 伸缩因子 η 作为微观调控因子加强算法的局部开发能力, 在基本透镜成像反向学习策略中, 伸缩因子一般作为固定值使用, 不利于算法在解空间进行全方位的探索. 本文提出一种基于非线性动态递减的伸缩因子策略, 在算法迭代前期可以获得较大值, 使算法可以在不同维度区域进行更大范围的搜索, 提高群体的多样性; 在算法迭代后期获得较小值, 在最优个体附近进行精细搜索, 提高局部寻优能力. 非线性动态伸缩因子计算公式为

$$\eta = \eta_{\max} - (\eta_{\max} - \eta_{\min}) \cdot (t/t_{\max})^2. \quad (12)$$

其中: η_{\max} 和 η_{\min} 分别为最大伸缩因子和最小伸缩因子, t_{\max} 为最大迭代次数.

将式(11)推广至 D 维搜索空间可得

$$X_j^* = \frac{ub_j + lb_j}{2} + \frac{ub_j + lb_j}{2 \cdot \eta} - \frac{X_j}{\eta}. \quad (13)$$

其中: X_j 和 X_j^* 分别为 X 和 X^* 的第 j 维向量, ub_j 和 lb_j 分别为决策变量的第 j 维向量. 通过对算法中全局最优个体进行凸透镜成像反向学习, 将各维度的值映射到空间中得到反向解, 不仅可避免各维度之间的

干扰,而且可以扩大算法的搜索范围.虽然凸透镜的反向学习策略极大程度地提高了算法的求解精度,在一定程度上帮助种群跳出局部最优,但是无法直接判断产生新的反向个体是否优于原始个体.因此,采用贪婪机制比较新旧个体适应度值,进而筛选出最优个体,通过这种方式不断获得更优解,提高算法的寻优能力.其中贪婪机制的数学模型描述如下:

$$X_{new}(t) = \begin{cases} X^*, f(X) \geq f(X^*); \\ X, f(X) < f(X^*). \end{cases} \quad (14)$$

2.3 水波动态自适应因子

在标准ChOA中,攻击者是种群的领导者,对于种群最终位置更新有着至关重要的作用.在全局探索过程中,攻击者引导种群探寻更广阔的搜索区域,一旦攻击者陷入小范围最优解时,种群将受其影响陷入局部极值空间,导致算法出现停止搜索现象.因此,本文在攻击者位置更新公式中加入水波动态自适应因子,其迭代1000次的值如图3所示.利用水波动态变化的不确定性使得攻击者能在更广泛的区域搜索,降低其他个体跟随的盲目性,增强群体间的信息交流和学习,保持种群多样性,有效避免了趋同性,进而提高算法跳出局部最优的能力.

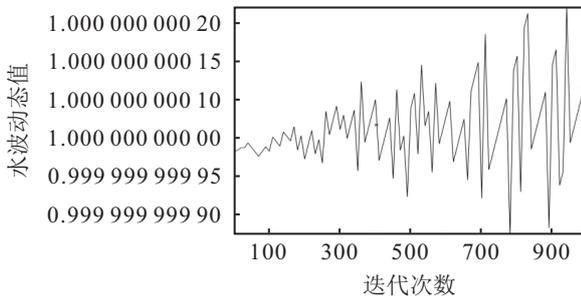


图3 迭代1000次水波动态值分布

水波动态自适应因子的数学模型为

$$\lambda = 1 - \sin\left(\frac{\pi \cdot t}{2 \cdot t_{max}} + 2 \cdot \pi\right). \quad (15)$$

ChOA引入水波动态自适应因子后攻击者位置更新公式为

$$X_1 = \lambda \cdot X_{attacker} - a_1 \cdot |C_1 \cdot X_{attacker} - m_1 \cdot X|. \quad (16)$$

2.4 SLWChOA实现步骤

综合上述改进方法,本文所提SLWChOA实现步骤如下.

step 1: 初始化算法相关参数:种群规模 N 、空间维度 dim 、种群的可搜索空间 $[ub, lb]$ 、最大迭代次数 t_{max} ;

step 2: 按照式(9)利用Sobol序列初始化种群;

step 3: 计算每个黑猩猩个体的适应度值,并选择

适应度最小的前4个个体位置,分别记录为 $X_{attacker}$ 、 $X_{barrier}$ 、 X_{chaser} 和 X_{driver} ;

step 4: 按照式(3)~(5)计算参数 a 、 m 、 C 的值;

step 5: 按照式(13)对 $X_{attacker}$ 进行凸透镜成像反向学习,产生反向解 $X_{attacker}^*$,根据式(14)贪婪机制选择适应度值低的解;

step 6: 按照式(7)和(16)更新 $X_{barrier}$ 、 X_{chaser} 、 X_{driver} 和 $X_{attacker}$,进一步根据式(8)更新黑猩猩种群位置;

step 7: 判断是否满足迭代终止条件,满足则输出全局最优黑猩猩位置 $X_{attacker}$,否则进入step 3继续执行.

2.5 改进算法时间复杂度分析

时间复杂度间接反映算法的收敛速度.在ChOA算法中,假设初始化参数(种群规模 N 、搜索空间维度 n 、系数 a 、 m 、 C 等参数)所需时间为 α_1 ,每一维按式(8)更新种群其他黑猩猩个体位置所需时间为 α_2 ,求解目标适应度函数时间为 $f(n)$,则ChOA的时间复杂度为

$$T_1(n) = O(\alpha_1 + N(n\alpha_2 + f(n))) = O(n + f(n)). \quad (17)$$

在SLWChOA算法中,初始化参数所需的时间与标准ChOA保持一致,采用Sobol序列初始化种群所需时间为 α_3 ,在算法循环阶段,设执行凸透镜成像反向学习策略所需时间为 α_4 ,执行贪婪机制时间为 α_5 ,每一维按照式(16)更新攻击者位置所需时间为 α_6 ,则SLWChOA的时间复杂度为

$$T_2(n) = O(\alpha_1 + \alpha_5 + N(n\alpha_3 + \alpha_4 + n\alpha_6 + f(n))) = O(n + f(n)). \quad (18)$$

本文所提SLWChOA与基本ChOA时间复杂度一致,即

$$T_1(n) = T_2(n) = O(n + f(n)). \quad (19)$$

综上所述,本文针对ChOA缺陷所提改进策略没有增加时间复杂度.

3 仿真实验与结果分析

3.1 实验设计与测试函数

本文基于Intel(R) Core(TM) i7-i7-6500U CPU、2.50 GHz主频、8 GB内存以及Windows 10(64位)的操作系统对所提出的算法进行仿真实验.编程软件为Matlab R2018a.表1为10个基准测试函数的基本信息,包含5个单峰函数 $F_1 \sim F_5$ 、3个非线性多峰函数 $F_6 \sim F_8$ 和2个固定维度的多峰函数 F_9 和 F_{10} .表2为每个对比算法的参数设置.

表1 基准测试函数

| 编号 | 名称 | 维度 | 定义域 | 理论值 |
|----------|-----------------|------------|-------------------|-----|
| F_1 | Sphere | 30/200/500 | $[-100, 100]$ | 0 |
| F_2 | Schwefel.2.21 | 30/200/500 | $[-100, 100]$ | 0 |
| F_3 | Rosenbrock | 30/200/500 | $[-30, 30]$ | 0 |
| F_4 | Step | 30/200/500 | $[-100, 100]$ | 0 |
| F_5 | Quartic | 30/200/500 | $[-1.28, 1.28]$ | 0 |
| F_6 | Rastrigin | 30/200/500 | $[-5.12, 5.12]$ | 0 |
| F_7 | Ackley | 30/200/500 | $[-32, 32]$ | 0 |
| F_8 | Griewank | 30/200/500 | $[-600, 600]$ | 0 |
| F_9 | Foxholes | 2 | $[-65.56, 65.56]$ | 1 |
| F_{10} | Goldstein-Price | 2 | $[-2, 2]$ | 3 |

表2 参数设置表

| 算法 | 主要参数 |
|---------|---|
| ChOA | $r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$ |
| SChOA | $r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$ |
| LChOA | $r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$ $\eta_{\max} = 10, \eta_{\min} = 1$ |
| WChOA | $r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$ |
| SLWChOA | $r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$ $\eta_{\max} = 10, \eta_{\min} = 1$ |

3.2 不同改进策略对算法性能影响分析

为充分验证SLWChOA的有效性和优越性,将基本ChOA与本文加入Sobol序列初始化的黑猩猩算法(SChOA)、加入凸透镜成像反向学习策略的黑猩猩算法(LChOA)、加入水波动态自适应因子的黑猩猩算法(WChOA)在10个具有不同寻优特征的基准测试函数上进行仿真实验对比. 算法通用条件设置相同,种群大小 $N = 30$,空间维度 $\dim = 30$,最大迭代次数 $t_{\max} = 1000$. 表3通过最优值、最差值、平均值和标准差4个性能指标评估每个算法的寻优性能. 5种算法对10个基准测试函数的求解结果如表3所示.

表3通过最优值和平均值反映算法的寻优性能和收敛精度的能力. 首先,就最优值而言,SLWChOA对 F_1 、 F_2 、 F_6 和 F_8 求解时,获得理论值. 对于函数 $F_3 \sim F_5$,其形状类似于抛物面,存在大量局部最优值,同时 F_7 是具有山谷状的多峰函数,其全局最优值位于山低端比较难寻,因此所提改进算法与其他对比算法均未寻到理论值,但是SChOA、LChOA和WChOA的寻优精度和稳定性优于标准ChOA,说明不同改进策略对算法的寻优性能有不同程度的提高. 其次,就平均值而言,仅采用Sobol序列初始化策略(SChOA)对ChOA性能的改进有限,而LChOA算法在函数 F_1 、 F_2 、 F_6 和 F_8 上均能寻到理论值,这是因为凸透镜成像的反向学习策略产生将反向解应用到当前最优解上,其引导种群向全局最优区域靠拢,提高了算法的寻优精度和收敛速度. 在固定维度的

表3 基准测试函数求解结果对比

| 函数 | 算法 | 最优值 | 最差值 | 平均值 | 标准差 |
|----------|---------|-----------------|-----------------|-----------------|-----------------|
| F_1 | ChOA | 1.52e-23 | 1.09e-12 | 6.18e-13 | 2.07e-13 |
| | SChOA | 2.81e-34 | 3.96e-13 | 2.09e-14 | 8.07e-14 |
| | LChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | WChOA | 7.44e-95 | 4.69e-89 | 2.91e-90 | 9.04e-90 |
| F_2 | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | ChOA | 1.52e-06 | 2.13e-02 | 2.44e-03 | 4.52e-03 |
| | SChOA | 2.52e-07 | 1.03e-02 | 1.65e-04 | 2.46e-04 |
| | LChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | WChOA | 8.56e-41 | 3.25e-37 | 3.36e-38 | 7.57e-38 |
| F_3 | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | ChOA | 2.81e+01 | 2.90e+01 | 2.88e+01 | 2.24e-01 |
| | SChOA | 2.87e+01 | 2.90e+01 | 2.89e+01 | 8.24e-02 |
| | LChOA | 2.72e+01 | 2.90e+01 | 2.78e+01 | 3.65e-01 |
| F_4 | WChOA | 2.72e+01 | 2.90e+01 | 2.80e+01 | 4.19e-01 |
| | SLWChOA | 2.63e+01 | 2.88e+01 | 2.68e+01 | 3.06e-01 |
| | ChOA | 2.38e+00 | 3.77e+00 | 3.20e+00 | 3.59e-01 |
| | SChOA | 2.04e+00 | 3.80e+00 | 3.10e+00 | 3.54e-01 |
| F_5 | LChOA | 1.56e+00 | 2.81e+00 | 2.33e+00 | 3.37e-01 |
| | WChOA | 2.54e+00 | 3.66e+00 | 2.67e+00 | 3.22e-01 |
| | SLWChOA | 1.04e+00 | 2.50e+00 | 1.37e+00 | 2.47e-01 |
| | ChOA | 5.19e-05 | 3.85e-03 | 7.76e-04 | 7.50e-04 |
| F_6 | SChOA | 9.89e-05 | 3.38e-03 | 1.05e-04 | 9.09e-04 |
| | LChOA | 9.86e-06 | 5.84e-04 | 2.10e-04 | 1.44e-04 |
| | WChOA | 6.93e-06 | 5.90e-04 | 1.01e-04 | 1.11e-04 |
| | SLWChOA | 1.74e-05 | 6.37e-04 | 1.71e-05 | 1.44e-05 |
| F_7 | ChOA | 4.89e-12 | 3.30e+01 | 3.85e+00 | 6.23e+00 |
| | SChOA | 4.88e-11 | 5.51e+00 | 1.69e+00 | 1.83e+00 |
| | LChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | WChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | SLWChOA | 8.88e-16 | 8.88e-16 | 8.88e-16 | 0.00e+00 |
| F_8 | ChOA | 2.00e+01 | 2.00e+01 | 2.00e+01 | 1.36e-03 |
| | SChOA | 3.72e-07 | 2.00e+01 | 1.85e+00 | 5.02e+00 |
| | LChOA | 8.88e-16 | 8.88e-16 | 8.88e-16 | 0.00e+00 |
| | WChOA | 2.00e+00 | 2.00e+00 | 2.00e+00 | 1.47e-03 |
| | SLWChOA | 8.88e-16 | 8.88e-16 | 8.88e-16 | 0.00e+00 |
| F_9 | ChOA | 2.22e-16 | 6.41e-02 | 1.53e-02 | 1.87e-02 |
| | SChOA | 0.00e+00 | 5.09e-02 | 8.02e-03 | 1.38e-03 |
| | LChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | WChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| F_{10} | ChOA | 9.98e-01 | 9.98e-01 | 9.98e-01 | 1.83e-05 |
| | SChOA | 9.98e-01 | 9.98e-01 | 9.98e-01 | 1.23e-05 |
| | LChOA | 9.98e-01 | 9.98e-01 | 9.98e-01 | 4.95e-05 |
| | WChOA | 9.98e-01 | 9.98e-01 | 9.98e-01 | 2.55e-05 |
| | SLWChOA | 9.99e-01 | 9.98e-01 | 9.99e-01 | 2.31e-05 |
| F_{10} | ChOA | 3.00e+00 | 3.00e+00 | 3.00e+00 | 9.38e-05 |
| | SChOA | 3.00e+00 | 3.00e+00 | 3.00e+00 | 9.30e-05 |
| | LChOA | 3.00e+00 | 3.00e+00 | 3.00e+00 | 6.30e-05 |
| | WChOA | 3.00e+00 | 3.00e+00 | 3.00e+00 | 7.99e-05 |
| | SLWChOA | 3.00e+00 | 3.00e+00 | 3.00e+00 | 5.62e-05 |

多峰函数 F_9 上,SLWChOA寻优精度达到最高,对于函数 F_{10} ,SChOA、LChOA、WChOA和SLWChOA均能寻到理论值,且SLWChOA的标准差表现排名第1,SChOA、WChOA、LChOA的标准差表现分别位居第

2~第4,均优于标准ChOA,表明SLWChOA稳定性和鲁棒性更好.综上所述,所提改进策略对标准ChOA具有明显的优化效果,融合3种策略的SLWChOA无论在寻优精度还是稳定性上均表现出一定的优势.

3.3 SLWChOA收敛性分析

为了反映SLWChOA的动态收敛特性,图4给出了4个基准测试函数的平均收敛曲线.

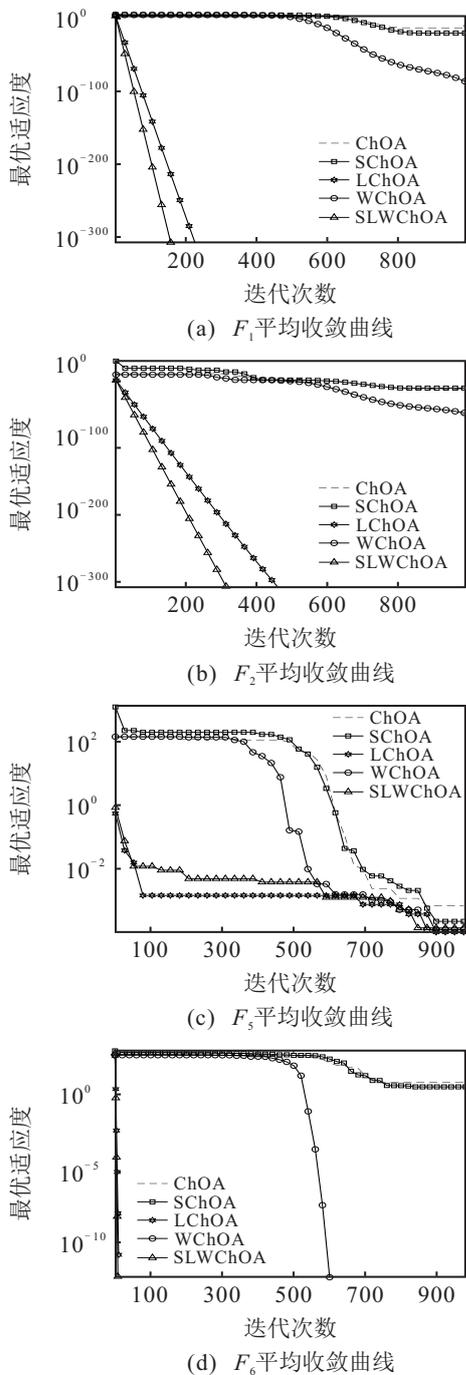


图4 不同策略算法平均收敛曲线

由图4可知,LChOA和WChOA求解单峰函数 F_1 、 F_2 和多峰函数 F_6 时,其寻优精度均比标准ChOA分别至少提高200个数量级和30个数量级,这归功

于凸透镜成像反向学习策略产生的更优解能够引导种群稳定寻到全局最优解,提高算法的收敛精度和速度,同时在水波动态自适应因子的协助下,“攻击者”个体带领种群向更大的猎物搜索范围寻找最优解,保持种群多样性的同时帮助算法跳出局部最优. SLWChOA求解函数 F_5 时陷入局部空间极值且难以跳出,同时其他对比算法也出现停滞寻优,但SLWChOA的平均收敛曲线均位于4种算法平均收敛曲线下,达到特定精度所需的迭代次数最少. 具体来说,SLWChOA在迭代前期的搜索性能和迭代末期的开拓性能都优于其他对比算法,在相同的迭代次数下具有更高的求解精度和更快的收敛速度,表明SLWChOA在保证开拓能力的同时也能充分保证探索能力,不失群体多样性和寻优稳定性.

综上,表3的实验结果和图4的平均曲线均验证了本文所提改进算法的有效性.虽然在某些函数上收敛精度差距不明显,但收敛速度远高于其他比较算法. SLWChOA的综合寻优能力比其他对比算法更强,稳定性更高.

3.4 SLWChOA与群智能算法及改进算法对比

为了更进一步验证其有效性,将SLWChOA与其他改进的蝴蝶优化算法MSWOA^[5](mixed strategy to improve butterfly optimization algorithm)、改进的灰狼优化算法LILGWO^[6](grey wolf optimizer algorithm based on lens imaging learning strategy)、最新改进黑猩猩优化算法SChOA^[7](sine-cosine chimp optimization algorithm)、EChOA^[8](an enhanced chimp optimization algorithm)以及经典的粒子群算法^[10](PSO)进行比较,引用文献[7-8]的实验数据,并复现文献[5-6, 10]的实验.为了体现比较的公平性,采用与文献相同的实验参数设置(种群规模为30,最大迭代次数为500),表4和表5列出了每个算法对不同类型的基准测试函数的优化结果,其中单峰函数和多峰函数搜索维度分别设置为30/200/500,每个算法独立运行30次,记录其平均值和标准差(“-”为文献缺失数据).

表4实验结果表明,SLWChOA在函数 F_1 、 F_2 、 F_6 和 F_8 上均寻到了理论最优值.在其他函数中表现不好的EChOA在 F_3 和 F_5 上的表现发生了变化,获得最高寻优精度,而SLWChOA的表现位居第2.在函数 F_5 和 F_7 中,大量局部极小点分布在其解空间中,算法对全局最优值的寻找有一定的困难,然而SLWChOW相较于其他改进算法具有更高的收敛精度和稳定性.当维度从30维上升到200维再上升到500维时,算法的求解精度和鲁棒性均有不同程度下降,这是因

表4 不同维度下各算法对单峰函数与多峰函数的求解结果比较

| 函数 | 算法 | 30 dim | | 200 dim | | 500 dim | |
|-------|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | | 平均值 | 标准差 | 平均值 | 标准差 | 平均值 | 标准差 |
| F_1 | PSO | 1.20e-04 | 1.81e-04 | 3.78e-06 | 3.35e-06 | 5.91e+03 | 4.79e+02 |
| | SChOA ^[7] | 1.57e+03 | 5.68e+03 | — | — | — | — |
| | EChOA ^[8] | 1.22e-36 | 6.33e-36 | — | — | — | — |
| | MSWOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | LILGWO | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| F_2 | PSO | 1.14e+00 | 2.39e-01 | 9.46e+01 | 4.12e+00 | 2.79e+01 | 1.65e+00 |
| | SChOA ^[7] | 1.22e+01 | 1.27e+00 | — | — | — | — |
| | EChOA ^[8] | 2.57e-16 | 8.96e-16 | — | — | — | — |
| | MSWOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | LILGWO | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| F_3 | PSO | 7.43e+01 | 4.66e+01 | 1.98e+02 | 2.83e-01 | 3.67e+03 | 3.26e+06 |
| | SChOA ^[7] | 5.95e+05 | 1.00e+07 | — | — | — | — |
| | EChOA ^[8] | 1.04e+01 | 8.21e-01 | — | — | — | — |
| | MSWOA | 2.86e+01 | 5.71e-01 | 1.98e+02 | 6.36e-01 | 8.80e+02 | 4.67e+00 |
| | LILGWO | 2.88e+01 | 8.82e-01 | 1.98e+02 | 5.37e-01 | 8.01e+02 | 1.70e+00 |
| | SLWChOA | 2.70e+01 | 3.17e-01 | 1.46e+02 | 5.78e-14 | 4.02e+02 | 2.89e-14 |
| F_4 | PSO | 2.51e-04 | 7.77e-04 | 6.10e+04 | 2.26e+03 | 6.01e+04 | 2.52e+03 |
| | SChOA ^[7] | 1.01e+03 | 5.62e+03 | — | — | — | — |
| | EChOA ^[8] | 1.10e+00 | 4.33e-01 | — | — | — | — |
| | MSWOA | 2.64e+00 | 3.47e-01 | 3.54e+01 | 1.94e+00 | 8.89e+01 | 3.57e+00 |
| | LILGWO | 2.66e+00 | 5.00e-01 | 3.33e+01 | 1.17e+00 | 1.01e+02 | 1.74e+00 |
| | SLWChOA | 1.63e+00 | 2.58e-01 | 3.14e+01 | 1.45e-14 | 8.00e+01 | 2.89e-14 |
| F_5 | PSO | 1.90e-01 | 7.80e-02 | 6.08e+04 | 2.20e+03 | 6.03e+04 | 1.72e+03 |
| | SChOA ^[7] | 1.44e+00 | 5.77e+00 | — | — | — | — |
| | EChOA ^[8] | 1.09e-06 | 7.99e-07 | — | — | — | — |
| | MSWOA | 5.99e-01 | 3.72e-01 | 1.20e+00 | 0.90e+00 | 6.69e-01 | 3.32e-01 |
| | LILGWO | 3.39e-04 | 3.46e-04 | 5.66e-04 | 6.54e-04 | 3.86e-04 | 2.86e-04 |
| | SLWChOA | 1.73e-05 | 1.21e-05 | 2.87e-05 | 2.18e-05 | 3.30e-05 | 2.82e-05 |
| F_6 | PSO | 5.53e+01 | 1.13e+01 | 6.02e+04 | 2.10e+03 | 5.92e+04 | 3.31e+03 |
| | SChOA ^[7] | 4.40e+01 | 7.77e+01 | — | — | — | — |
| | EChOA ^[8] | 1.41e-01 | 6.05e-01 | — | — | — | — |
| | MSWOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | LILGWO | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| F_7 | PSO | 4.06e-01 | 6.37e-01 | 5.97e+04 | 2.82e+03 | 5.94e+04 | 2.08e+03 |
| | SChOA ^[7] | 1.89e+00 | 1.71e+00 | — | — | — | — |
| | EChOA ^[8] | 1.94e-14 | 3.63e-15 | — | — | — | — |
| | MSWOA | 8.88e-16 | 0.00e+00 | 8.88e-16 | 0.00e+00 | 8.88e-16 | 0.00e+00 |
| | LILGWO | 8.88e-16 | 0.00e+00 | 8.88e-16 | 0.00e+00 | 8.88e-16 | 0.00e+00 |
| | SLWChOA | 8.88e-16 | 0.00e+00 | 8.88e-16 | 0.00e+00 | 8.88e-16 | 0.00e+00 |
| F_8 | PSO | 8.00e-03 | 8.32e-03 | 6.03e+04 | 3.06e+03 | 6.01e+04 | 2.13e+03 |
| | SChOA ^[7] | 2.09e+01 | 8.30e+01 | — | — | — | — |
| | EChOA ^[8] | 3.10e-03 | 7.33e-03 | — | — | — | — |
| | MSWOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | LILGWO | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |
| | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 |

表5 各算法求解固定维度多峰函数的结果比较 (dim = 2)

| 函数 | 算法 | 平均值 | 标准差 |
|----------|----------------------|-----------------|-----------------|
| F_9 | PSO | 2.61e+00 | 2.04e+00 |
| | SChOA ^[7] | 1.33e+00 | 1.04e+01 |
| | EChOA ^[8] | — | — |
| | MSWOA | 1.79e+00 | 1.90e+00 |
| | LILGWO | 7.60e+00 | 4.73e+00 |
| | SLWChOA | 9.99e-01 | 2.67e-02 |
| F_{10} | PSO | 3.00e+00 | 1.47e-05 |
| | SChOA ^[7] | 3.00e+00 | 1.92e-01s |
| | EChOA ^[8] | — | — |
| | MSWOA | 3.00e+00 | 2.48e-05 |
| | LILGWO | 3.00e+00 | 2.67e-05 |
| | SLWChOA | 3.00e+00 | 1.43e-05 |

为随着维度增加,算法搜索范围不断扩大,寻优过程需要更多计算,但是相较于5种对比算法,SLWChOA求解精度在大部分函数上仍最高.因此,SLWChOA在求解低维和高维问题时具有极强的寻优性能和鲁棒性,进一步说明了SLWChOA在求解复杂函数优化问题时具有显著的竞争优势.

由表5可知,SLWChOA在2个固定维度的多峰函数上寻优精度都达到了最高.在函数 F_9 上,SLWChOA获得最高寻优精度且标准差最低,PSO、LILGWO和SLWChOA在函数 F_{10} 上收敛到理论最优值,但SLWChOA的标准差最小,说明SLWChOA具有

更好的寻优稳定性.

3.5 改进算法与其他算法在不同维度上比较

本节基于10个基准测试函数,使用平均绝对误差(mean absolute error, MAE)^[11]评估各算法性能差异的真实情况,维数选取30~510维,呈等差数列递增120维变化.通过MAE对算法进行排序,更进一步地验证算法的性能指标,避免因最优值近似而引起算法趋同性.实验参数设置种群规模为30,最大迭代次数为500,各算法独立运行30次.表6为基准测试函数的MAE排序结果,MAE的数学模型如下:

$$MAE = \sum_{i=1}^{N_f} |\vartheta_i - \delta_i| / N_f. \quad (20)$$

其中: ϑ_i 为算法最优值的平均值, δ_i 为每个基准函数的理论最优值, N_f 为所选用基准函数的个数.

由表6可知,在30维中,EChOA的MAE最小,而SLWChOA的MAE表现位列第2.随着维度的不断增加,SLWChOA的MAE排名保持第一,说明SLWChOA的寻优性能不会随着复杂高维多极值的维度增加而陷入“维数灾难”,相较于其他对比算法具有更显著优势,验证了所提算法的有效性.

表6 MAE算法排名

| 算法 | 30 dim | | 150 dim | | 270 dim | | 390 dim | | 510 dim | |
|---------|--------------|----------|---------------|----------|---------------|----------|---------------|----------|---------------|----------|
| | MAE | 排名 | MAE | 排名 | MAE | 排名 | MAE | 排名 | MAE | 排名 |
| PSO | 13.295 | 5 | 3.574e+04 | 4 | 2.449e+05 | 4 | 3.735e+05 | 4 | 3.297e+06 | 4 |
| SChOA | 5.97e+04 | 6 | — | — | — | — | — | — | — | — |
| EChOA | 1.455 | 1 | — | — | — | — | — | — | — | — |
| MSWOA | 3.262 | 3 | 17.453 | 2 | 31.674 | 2 | 35.015 | 2 | 99.859 | 2 |
| LILGWO | 3.806 | 4 | 17.910 | 3 | 32.340 | 3 | 35.830 | 3 | 101.560 | 3 |
| SLWChOA | 2.862 | 2 | 13.170 | 1 | 22.120 | 1 | 32.220 | 1 | 53.750 | 1 |

3.6 改进算法在高维上的性能分析

本节选取与3.5节相同变化的维度,限于篇幅,选用4个不同类型的函数(单峰函数 F_1 、 F_5 和多峰函数 F_6 、 F_7),将算法优化函数的平均值及其平均变化率作

为评价指标,测试本文算法SLWChOA相较于ChOA独立运行30次的性能随函数维度的增加而变化的情况,设置种群规模为30,最大迭代次数为500,结果如表7所示.平均变化率的数学模型为

$$\text{平均变化率} = \left(\sum \frac{\text{后一次维度的平均值} - \text{前一次维度的平均值}}{\text{前一次维度的平均值}} \right) / 4. \quad (21)$$

表7 SLWChOA与ChOA在不同维度上优化函数平均值比较

| 函数 | 算法 | 维度 | | | | | 平均变化率/% |
|-------|---------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | | 30 | 150 | 270 | 390 | 510 | |
| F_1 | ChOA | 7.71e-06 | 8.08e+00 | 9.54e+01 | 3.34e+02 | 7.73e+02 | 2.61e+05 |
| | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0 |
| F_5 | ChOA | 1.47e-03 | 5.91e-02 | 3.30e-01 | 1.19e+00 | 3.11e+00 | 1.23e+01 |
| | SLWChOA | 1.71e-05 | 2.76e-05 | 2.89e-05 | 3.42e-05 | 3.53e-05 | 2.19e-01 |
| F_6 | ChOA | 1.42e+00 | 5.38e+01 | 1.41e+02 | 2.21e+02 | 3.33e+02 | 9.90e+00 |
| | SLWChOA | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0.00e+00 | 0 |
| F_7 | ChOA | 1.98e+01 | 2.00e+01 | 2.00e+01 | 2.01e+01 | 2.01e+01 | 3.77e-03 |
| | SLWChOA | 8.88e-16 | 8.88e-16 | 8.88e-16 | 8.88e-16 | 8.88e-16 | 0 |

由表7可知,随着维度的增加,SLWChOA的平均值基本保持不变.具体而言,对于函数 F_1 、 F_6 ,SLWChOA的平均值均能保持为0,然而ChOA随着维度的递增,平均值呈现不断增加趋势.在被大量局部极小值包围的函数 F_5 上,SLWChOA的平均值基本保持不变,而ChOA的平均值变化幅度略大于SLWChOA.在函数 F_1 、 F_6 和 F_7 上,SLWChOA的平均变化率均为0,远低于ChOA的平均变化率.

图5是两种算法在不同维度下的平均值变化曲线.从图5可以观察到,在函数 F_1 、 F_5 和 F_6 上,ChOA的平均值随维度递增而大幅度增加,而SLWChOA的平均值保持不变.这说明维度的不断增加,对SLWChOA的寻优能力影响较小,相比于ChOA具有更突出的寻优性能,进一步验证了本文所提算法的优越性.

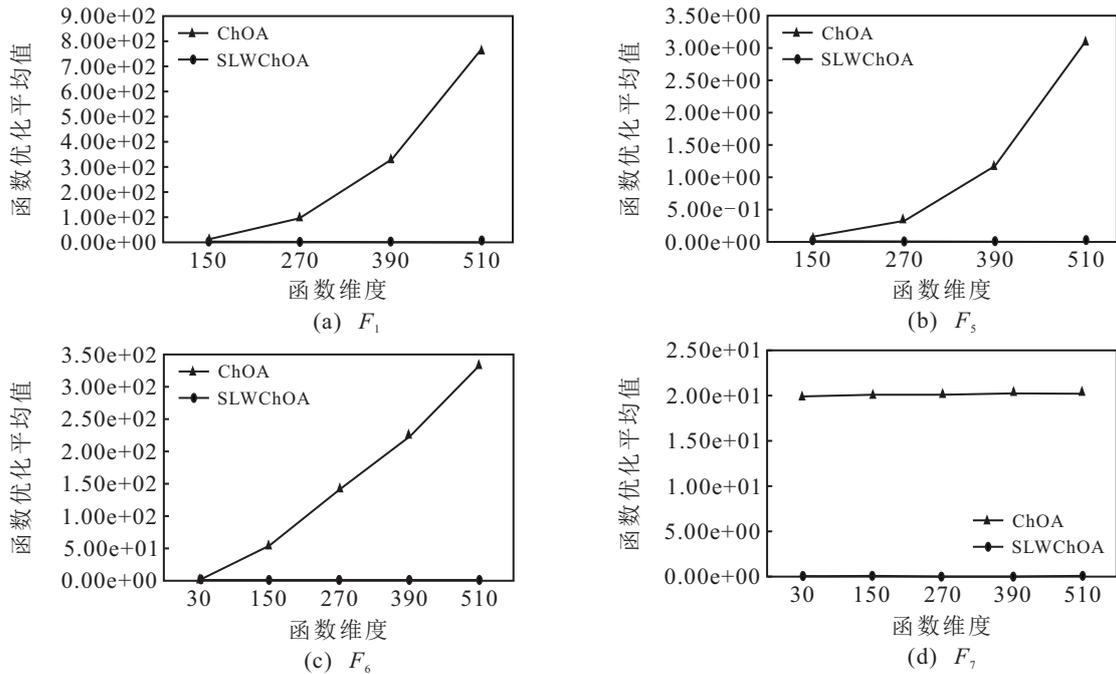


图5 SLWChOA和ChOA优化函数平均值随函数维度变化

3.7 Wilcoxon秩和检验

为了体现改进算法的有效性,在显著性水平 $p = 5\%$ 、维度为30的情况下,本文采用Wilcoxon秩和检验验证SLWChOA是否在统计上与标准ChOA、SChOA、LChOA、WChOA及新提出的元启发式算法(AOA)^[12]和经典PSO算法存在明显差异.符号“+”

“-”“=”分别表示SLWChOA性能优于、劣于和相当于对比算法,“NaN”表示不适用,即无法进行显著性判断,结果如表8所示,在10个基准测试函数中,大部分 p 值都小于5%,总体上SLWChOA的性能与其他6种算法在统计上具有显著性差异,从而表明SLWChOA比其他算法拥有更好的有效性.

表8 Wilcoxon秩和检验结果

| 函数 | ChOA (p_1) | SChOA (p_2) | LChOA (p_3) | WChOA (p_4) | PSO (p_5) | AOA (p_6) |
|-----------|----------------|-----------------|-----------------|-----------------|---------------|---------------|
| F_1 | 3.31e-20 | 3.31e-20 | 3.31e-20 | 3.31e-20 | 1.84e-10 | 1.84e-10 |
| F_2 | 3.31e-20 | 3.31e-20 | 3.31e-20 | 3.31e-20 | 1.84e-10 | 1.84e-10 |
| F_3 | 3.30e-02 | 3.40e-02 | 3.87e-02 | 4.29e-02 | 4.04e-02 | 2.50e-05 |
| F_4 | 4.92e-02 | 4.39e-02 | 9.00e-02 | 1.54e-02 | 1.44e-02 | 6.32e-12 |
| F_5 | 3.36e-02 | 9.97e-02 | 1.63e-02 | 8.93e-02 | 7.07e-18 | 7.05e-07 |
| F_6 | NaN | NaN | NaN | NaN | 1.84e-10 | NaN |
| F_7 | 3.30e-20 | 3.08e-20 | 2.98e-20 | 3.31e-20 | 1.84e-10 | 2.70e-09 |
| F_8 | NaN | NaN | NaN | NaN | 1.84e-10 | NaN |
| F_9 | 1.81e-02 | 2.46e-02 | 3.87e-02 | 4.29e-02 | 1.95e-03 | 2.50e-05 |
| F_{10} | 1.17e-02 | 1.77e-03 | 5.08e-03 | 1.99e-02 | 5.60e-07 | 2.35e-11 |
| + / = / - | 8/2/0 | 7/2/1 | 7/2/1 | 7/2/1 | 10/0/0 | 8/2/0 |

3.8 SLWChOA求解CEC2014测试函数

为更好地验证SLWChOA的鲁棒性,本文采用SLWChOA、基本ChOA、PSO算法、MSBOA、LILGWO算法以及L-SHADE(linear-success-history based adaptive differential evolution)算法^[13]对部分具有复杂特征的CEC2014测试函数进行优化实验,其中包含单峰(CEC03)、多峰(CEC5、CEC6、CEC16)、混合(CEC19)和复合(CEC23)类型的函数,选取的部分函数具体信息如表9所示.为保证对比的公平性,最大迭代次数设置为1000,每个算法分别独立运行30次,PSO算法和L-SHADE算法的数据来源于文献[14],结果如表10所示.

表9 CEC2014测试函数(部分)

| 函数 | 维度 | 特征 | 定义域 | 最优值 |
|-------|----|----|-------------|------|
| CEC03 | 30 | UN | [-100, 100] | 300 |
| CEC5 | 30 | MF | [-100, 100] | 500 |
| CEC6 | 30 | MF | [-100, 100] | 600 |
| CEC16 | 30 | MF | [-100, 100] | 1600 |
| CEC19 | 30 | HF | [-100, 100] | 1900 |
| CEC23 | 30 | CF | [-100, 100] | 2300 |

由表10可知,L-SHADE在单峰函数CEC03上表现出色,SLWChOA收敛精度低于L-SHADE,因为SLWChOA需要进行更多的参数计算,造成收敛精度有所下降,而SLWChOA的寻优性能略优于标准ChOA.在多峰(CEC5、CEC6、CEC16)和混合函数

表 10 CEC2014测试函数优化结果对比

| 算法 | 指标 | CEC03 | CEC5 | CEC6 | CEC16 | CEC19 | CEC23 |
|-------------------------|-----|-------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| ChOA | 平均值 | 8.526 2e+04 | 5.210 4e+02 | 6.395 5e+02 | 1.613 2e+03 | 2.267 7e+03 | 2.753 0e+03 |
| | 标准差 | 4.577 9e+03 | 5.437 4e-02 | 2.265 5e+00 | 2.227 8e-01 | 1.173 6e+02 | 6.158 4e+01 |
| PSO ^[14] | 平均值 | 4.87e+01 | 2.00e+01 | 1.00e+01 | 1.13e+01 | 7.76e+00 | 3.16e+02 |
| | 标准差 | 6.61e+01 | 2.79e-02 | 2.25e+00 | 7.05e-01 | 1.87e+00 | 3.52e-01 |
| LILGWO | 平均值 | 4.789 7e+04 | 5.210 5e+02 | 6.199 4e+02 | 1.612 9e+03 | 1.985 9e+03 | 2.500 0e+03 |
| | 标准差 | 1.069 8e+04 | 4.733 2e-02 | 2.767 2e+00 | 5.402 3e-01 | 3.082 5e+01 | 0.000 0e+00 |
| MSBOA | 平均值 | 4.462 0e+04 | 5.217 3e+02 | 6.343 5e+02 | 1.612 6e+03 | 1.943 8e+03 | 2.500 0e+03 |
| | 标准差 | 8.065 2e+03 | 9.529 4e-02 | 2.901 6e+00 | 3.826 4e-01 | 1.424 4e+01 | 0.000 0e+00 |
| L-SHADE ^[14] | 平均值 | 0.00e+00 | 2.01e+01 | 1.67e-02 | 8.48e+00 | 3.121e+03 | 3.15e+02 |
| | 标准差 | 0.00e+00 | 1.70e-02 | 9.17e-02 | 2.97e-01 | 7.2e-01 | 1.14e-13 |
| SLWChOA | 平均值 | 7.985 4e+04 | 5.210 3e+02 | 6.374 4e+02 | 1.612 0e+03 | 2.150 8e+03 | 2.500 0e+03 |
| | 标准差 | 7.750 6e+03 | 5.530 6e-02 | 2.369 0e+00 | 3.011 3e-01 | 1.147 9e+02 | 0.000 0e+00 |

(CEC19)上,SLWChOA的寻优精度更加接近理论值且比ChOA稳定.在复合函数上,SLWChOA的收敛精度同LILGWO算法、MSBOA一致,远高于其他3个对比算法,验证了SLWChOA具有较好的寻优鲁棒性.

4 机械优化设计问题

优化问题作为工程设计和应用领域中经常出现的数值约束问题,传统的机械方法难以解决非线性甚至高维的数值优化问题.区别于传统方法,本文将所提SLWChOA用于优化伸张/压缩弹簧机械设计问题,进一步验证改进算法的可行性和适用性.

4.1 伸张/压缩弹簧设计优化设计案例

伸张/压缩弹簧设计问题的优化目标是减少弹簧的重量,其结构示意图如图6所示.

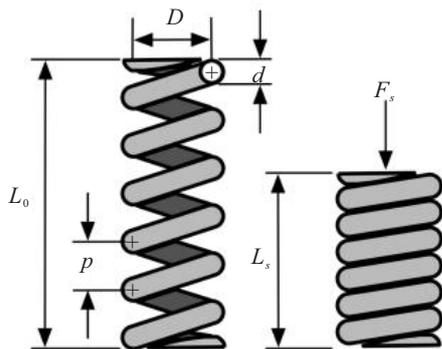


图 6 伸张/压缩弹簧结构

约束条件包括受到最小偏差(g_1)、剪切应力(g_2)、冲击频率(g_3)、外径限制(g_4),决策变量包括线径 d 、平均线圈直径 D 及有效线圈数 P , $f(x)$ 为最小化弹簧重量.伸张/压缩弹簧设计的数学模型描述如下.

目标函数为

$$x = [x_1, x_2, x_3] = [d, D, N];$$

$$\min f(x) = x_1^2 x_2 (2 + x_3). \quad (22)$$

约束条件为

$$\begin{cases} g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0, \\ g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\ g_3(x) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0, \\ g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0. \end{cases} \quad (23)$$

其中: $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$.

4.2 测试结果与分析

本文将所提SLWChOA与黑猩猩优化算法(ChOA)、粒子群算法PSO、正余弦算法(sine cosine algorithm, SCA)^[15]、 Harris鹰优化算法(Harris hawks optimization, HHO)^[16]、均衡器优化算法(equilibrium optimization algorithm, EO)^[17]、斑鬣狗优化算法(spotted hyena optimization, SHO)^[18]、灰狼优化算法(grey wolf optimization, GWO)^[19]、多元宇宙优化算法(multi-verse optimization, MVO)^[20]、麻雀搜索算法(sparrow search algorithm, SSA)^[21]进行实验比较,其中对比算法的数据来源于文献[12, 22].实验中:种群规模为30,最大迭代次数为1000,每个算法独立运行30次并取平均值.表11分别是各算法伸张/压缩弹簧设计的优化结果.

表 11 伸张/压缩弹簧设计问题中各算法的最优解

| 算法 | d | D | N | $f(x)$ |
|---------------------|---------------|---------------|---------------|---------------|
| SCA | 0.0500 | 0.3168 | 14.1109 | 0.0132 |
| HHO ^[12] | 0.0562 | 0.4754 | 6.6670 | 0.0130 |
| EO ^[12] | 0.0512 | 0.3445 | 12.0455 | 0.0127 |
| SHO ^[22] | 0.0511 | 0.03437 | 12.0955 | 0.0127 |
| GWO ^[22] | 0.0500 | 0.3415 | 12.0734 | 0.0127 |
| MVO ^[22] | 0.0500 | 0.3159 | 14.2262 | 0.0128 |
| PSO ^[22] | 0.0500 | 0.3104 | 15.0000 | 0.0131 |
| SSA | 0.0551 | 0.4451 | 7.5125 | 0.0132 |
| ChOA | 0.0500 | 0.3146 | 14.4315 | 0.0134 |
| SLWChOA | 0.0612 | 0.6260 | 4.1894 | 0.0127 |

表11显示了SLWChOA与其他算法处理的约束函数值,SLWChOA获得的函数最优解为 $[x_1, x_2, x_3] = [0.0612, 0.06260, 4.1894]$,最优值 $f(x) = 0.0127$,表明SLWChOA在减少弹簧的重量上获得了最佳解决方案,进一步验证了SLWChOA在实际应用中的可行性和适用性。

5 结论

为改善ChOA的寻优性能,本文首先采用Sobol序列使初始化种群分布均匀,增加种群的多样性;其次,对种群在局部阶段聚集程度进行分析,引入基于凸透镜成像的反向学习策略生成动态的反向解,有效地提高算法的搜索精度和收敛速度;同时,在“攻击者”个体位置更新处加入水波动态自适应因子,引导种群向全局最优的狩猎搜索范围进行全方位搜索,保持种群寻优多样性的同时有效避免了趋同性,降低算法跳出局部最优的概率。通过对10个基准测试函数及其Wilcoxon检验及部分CEC2014函数等实验验证了改进算法具有更高的寻优性能和更强的鲁棒性。最后,对机械设计问题进行优化测试分析,验证了SLWChOA在实际工程问题中的适用性,为解决复杂的工程优化问题提供了一条新途径。

参考文献(References)

- [1] Sooncharoen S, Pongcharoen P, Hicks C. Grey wolf production scheduling for the capital goods industry[J]. *Applied Soft Computing*, 2020, 94: 106480.
- [2] Yan X A, Liu Y, Jia M P. A fault diagnosis approach for rolling bearing integrated SGMD, IMSDE and multiclass relevance vector machine[J]. *Sensors: Basel, Switzerland*, 2020, 20(15): 4352.
- [3] Gai J B, Shen J X, Hu Y F, et al. An integrated method based on hybrid grey wolf optimizer improved variational mode decomposition and deep neural network for fault diagnosis of rolling bearing[J]. *Measurement*, 2020, 162: 107901.
- [4] Khishe M, Mosavi M R. Chimp optimization algorithm[J]. *Expert Systems With Applications*, 2020, 149: 113338.
- [5] Ning J Q, He Q. Mixed strategy to improve butterfly optimization algorithm[J]. *Application Research of Computers*, 2021, 38(6): 1718-1723.
- [6] Long W, Wu T B, Tang M Z, et al. Grey wolf optimizer algorithm based on lens imaging learning strategy[J]. *Acta Automatica Sinica*, 2020, 46(10): 2148-2164.
- [7] Kaur M, Kaur R, Singh N, et al. SChOA: A newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications[J]. *Engineering With Computers*, 2021: 1-29.
- [8] Jia H M, Sun K J, Zhang W Y, et al. An enhanced chimp optimization algorithm for continuous optimization domains[J]. *Complex & Intelligent Systems*, 2022, 8(1): 65-82.
- [9] Bratley P, Fox B L. Implementing sobols quasirandom sequence generator (algorithm 659)[J]. *ACM Transactions on Mathematical Software*, 2003, 29(1): 49-57.
- [10] Kennedy J, Eberhart R. Particle swarm optimization[C]. *Proceedings of ICNN'95-International Conference on Neural Networks*. Perth, 1995: 1942-1948.
- [11] Nabil E. A modified flower pollination algorithm for global optimization[J]. *Expert Systems With Applications*, 2016, 57: 192-203.
- [12] Hashim F A, Hussain K, Houssein E H, et al. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems[J]. *Applied Intelligence*, 2021, 51(3): 1531-1551.
- [13] Tanabe R, Fukunaga A S. Improving the search performance of SHADE using linear population size reduction[C]. *2014 IEEE Congress on Evolutionary Computation*. Beijing, 2014: 1658-1665.
- [14] Xu G P. Study on particle swarm optimization based with dimensional learning strategy[D]. Changchun: Jilin University, 2019.
- [15] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems[J]. *Knowledge-Based Systems*, 2016, 96: 120-133.
- [16] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications[J]. *Future Generation Computer Systems*, 2019, 97: 849-872.
- [17] Faramarzi A, Heidarinejad M, Stephens B, et al. Equilibrium optimizer: A novel optimization algorithm[J]. *Knowledge-Based Systems*, 2020, 191: 105190.
- [18] Dhiman G, Kumar V. Spotted hyena optimizer for solving complex and non-linear constrained engineering problems[C]. *Harmony Search and Nature Inspired Optimization Algorithms*. Singapore, 2019: 857-867.
- [19] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [20] Mirjalili S, Mirjalili S M, Hatamlou A. Multi-verse optimizer: A nature-inspired algorithm for global optimization[J]. *Neural Computing and Applications*, 2016, 27(2): 495-513.
- [21] Xue J K, Shen B. A novel swarm intelligence optimization approach: Sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22-34.
- [22] Kaur S, Awasthi L K, Sangal A L, et al. Tunicate sarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization[J]. *Engineering Applications of Artificial Intelligence*, 2020, 90: 103541.

作者简介

何庆(1982—),男,副教授,博士,从事大数据应用、进化计算、自然语言处理等研究, E-mail: qhe@gzu.edu.cn;

罗仕杭(1998—),男,硕士生,从事群智能优化算法、神经网络的研究, E-mail: 1065678104@qq.com.

(责任编辑: 闫妍)